From Skiena and Revilla, Programming Challenges: *The Programming Contest Training Manual*
Springer-Verlag, New York, 2003. ISBN: 0-387-00163-8

110106 Interpreter

A certain computer has ten registers and 1,000 words of RAM. Each register or RAM location holds a three-digit integer between 0 and 999. Instructions are encoded as three-digit integers and stored in RAM. The encodings are as follows:

100	means *halt*
2*dn*	means *set register d to n (between 0 and 9)*
3*dn*	means *add n to register d*
4*dn*	means *multiply register d by n*
5*ds*	means *set register d to the value of register s*
6*ds*	means *add the value of register s to register d*
7*ds*	means *multiply register d by the value of register s*
8*da*	means *set register d to the value in RAM whose address is in register a*
9*sa*	means *set the value in RAM whose address is in register a to that of register s*
0*ds*	means *goto the location in register d unless register s contains 0*

All registers initially contain 000. The initial content of the RAM is read from standard input. The first instruction to be executed is at RAM address 0. All results are reduced modulo 1,000.

Input

The input begins with a single positive integer on a line by itself indicating the number of cases, each described as below. This is followed by a blank line, and there will be a blank line between each two consecutive inputs.

Each input case consists of up to 1,000 three-digit unsigned integers, representing the contents of consecutive RAM locations starting at 0. Unspecified RAM locations are initialized to 000.

Output

The output of each test case is a single integer: the number of instructions executed up to and including the *halt* instruction. You may assume that the program does halt. Separate the output of two consecutive cases by a blank line.

Sample Input

```
1

299
492
495
399
492
495
399
283
279
689
078
100
000
```

```
000
000
```

Sample Output

```
16
```